Brazilian Journal of Geophysics (2020) 38(3): 1–11
© 2020Brazilian Geophysical Society
ISSN 0102-261X
DOI 10.22564/rbgf.v38i3.2059

USING ADAPTIVE DIFFERENTIAL EVOLUTION ALGORITHM TO IMPROVE PARAMETER ESTIMATION IN SEISMIC PROCESSING: EXTENDED RESULTS

José Ribeiro, Tiago A. Coimbra, Nicholas T. Okita,
Gustavo B. Ignácio, and Martin Tygel

ABSTRACT. Since the end of the 1990s, methods of imaging and inversion have been receiving systematic attention, through multiparametric traveltimes, such as the Common-Reflection-Surface (CRS) method, in its two versions zero offset (ZO) and finite offset (FO), and the Offset Continuation Trajectory (OCT). Despite its superior quality to traditional methods, OCT and CRS face the challenges of additional computation costs, which stem from the required multiparameter estimations. The problem of estimating the slope, curvature, and velocity parameters reliably and efficiently has been drawing focus in the seismic literature. Mathematically, approaches to solve that problem rely on global optimization techniques. The main challenges are robustness (small relative sensitivity to given initial values) and convergence speed. The Differential Evolution (DE) has shown promising results. That method has a welcome property of robustness, however also the drawback of undesired convergence speed. In this paper, we propose overcoming this problem upon applying the Adaptive Differential Evolution known as JADE. Qualitative results from synthetic and real datasets show, for similar execution times, the fast convergence of JADE when compared with that of DE. Therefore, JADE presents itself as a great alternative to DE, showing even more promising results regarding estimating the parameters of OCT and CRS.

Keywords: seismic parameter estimation, adaptive differential evolution.

RESUMO. Desde o fim da década de 90, métodos de imageamento e inversão vêm recebendo atenção sistemática através de tempos de trânsito multiparamétricos, tal como o método Common-Reflection-Surface (CRS), em suas duas versões afastamento nulo (ZO) e afastamento finito (FO), e a trajetória para continuação de afastamento (OCT). Apesar da qualidade superior quando comparado aos métodos tradicionais, o OCT e o CRS enfrentam os desafios de custos adicionais de computação, que impedem a estimação dos parâmetros necesários. O problema de estimar os parâmetros de inclinação, curvatura e velocidade de uma maneira confiável e eficiente tem atraído o foco da literatura sísmica. Matematicamente, as abordagens para resolver esse problema contam com técnicas de otimização global. Os principais desafios são robustez (pequena sensibilidade relativa a determinados valores iniciais) e velocidade de convergência. Differential Evolution (DE) mostrou resultados promissores. Esse método possui uma propriedade bem-vinda de robustez, mas também a desvantagem de velocidade de convergência indesejada. Neste artigo, propomos superar esse problema com a aplicação do Adaptive Differential Evolution, conhecido como JADE. Os resultados qualitativos dos conjuntos de dados sintéticos e reais mostram, para tempos de execução semelhantes, a rápida convergência do JADE quando comparado ao do DE. Portanto, o JADE se apresenta como uma ótima alternativa ao DE, apresentando resultados ainda mais promissores em relação à estimação dos parâmetros do OCT e do CRS.

Palavras-chave: estimação de parâmetros sísmicos, evolução diferencial adaptativa.

Corresponding author: José Ribeiro

Universidade Estadual de Campinas, Centro de Estudos de Petróleo (CEPETRO/UNICAMP), Rua Cora Coralina, 350 - Cidade Universitária, Campinas - SP, Brazil, 13083-896 – E-mails: joseribeiro@ggaunicamp.com, tgo.coimbra@gmail.com, nicholas.okita@gmail.com, igna-ciogbd@ggaunicamp.com, mtygel@gmail.com

INTRODUCTION

In seismic data processing, reliable and accurate traveltime approximations to reflection and diffraction events play an essential role as stacking operators to generate initial sections or volumes from the given data. In general, a finite number of parameters define such traveltime operators. For each given data sample, a corresponding set of parameters must be estimated by some coherency analysis directly applied to the given dataset. More specifically, given a sample from the seismic data and a set of parameters, the corresponding traveltime must be constructed, and the data must be stacked along that traveltime. To compute this stacking's energy, that is, the level of coherence obtained with the given set, a semblance measure is used. In such a way, semblance plays the role of the objective function, and the set of parameters that maximizes it in the respective sample is the set desired. We refrain from providing the mathematical expression of semblance, but for a more detailed explanation, the reader is referred to, e.g., Neidell and Taner (1971).

In this paper, three traveltime operators are analyzed through parameter-estimation experiments. These are the Offset Continuation Trajectory (OCT) and the two hyperbolic traveltimes Zero Offset - Common Reflection Surface (ZO-CRS) and Finite Offset - Common Reflection Surface (FO-CRS). We also refrain from providing the mathematical expressions of OCT, ZO-CRS, and FO-CRS operators, referring the reader to the publications Coimbra et al. (2016), Jäger et al. (2001) and Zhang et al. (2001), respectively. For each data sample, ZO-CRS depends on three parameters (being one slope and two curvatures), FO-CRS on five parameters (two slopes and three curvatures), and OCT on two parameters (one slope and one velocity).

Parameter estimation

Concerning the parameter-estimation problem for the 2D ZO-CRS, which involves three parameters, Jäger et al. (2001) proposed a scheme that consists of a sequence of one-parameter searches in suitable sub-domains of the data, followed by a local, three-parameter search having the previously-obtained parameters as initial values. A generalization of that scheme has been proposed in Garabito et al. (2001), which is based on an auxiliary two-parameter (diffraction) traveltime operator. At first, the simulated annealing (SA) method is applied to provide a simultaneous estimate of these two parameters, upon which a (partial) stacked section is constructed. Next, an one-parameter search is applied to that partial stacked section to estimate the third parameter. In the same way as before, a local three-parameter search is carried out using the previous estimations as initial values.

From basic mathematical principles, sequential and few-parameter (cheaper) searches in sub-domains, followed by a final full-parameter local search are expected to provide less accurate estimations than that simultaneous, full-parameter, global (much more expensive) searches. Bonomi et al., 2009 applied the global conjugate direction optimization method (see Powell, 1964) to simultaneously estimate the eight parameters that define the 3D ZO-CRS operator. Estimations based on the global Differential Evolution (DE) method have been recently reported in (Barros et al., 2015, 2019) and Walda and Gajewski (2017).

In this work, we address the parameter-estimation problem on given 2D datasets, using for that the OCT, ZO-CRS, and FO-CRS traveltime operators. Our approach consists of applying the Adaptive Differential Evolution (see, e.g. Zhang and Sanderson, 2009), referred to as JADE, which combines the excellent accuracy and avoids the drawback of slow convergence provided by DE algorithm, as following shown in the results section. This paper is also an extension of the expanded abstract presented in the sixteenth International Congress of the Brazilian Geophysical Society (Ribeiro et al., 2019).

THEORETICAL REVISION

The methodology employed in our parameter estimation problem consists of the use of two global optimization metaheuristics, namely *Differential Evolution (DE)* and its counter-part *Adaptive Differential Evolution (JADE)*. Also, a graphics processing unit (GPU) implementation approach was followed to obtain better results in fewer amounts of time.

Differential Evolution (DE):

Differential Evolution (Storn and Price (1997)), or simply DE, is a metaheuristic (i.e., a global optimization method with no mathematical proof of convergence) that tries to formulate the optimization problem as an evolutionary process, in such a way that an initial population $P[0]$ (made up by solution-candidate individuals) is subjugated to an iterative process (with Ng iterations), in order to obtain a better population $P[Ng]$ at the Ng-th iteration.

For this initial population $P[0]$, Np individuals must be initialized, each one a D-dimensional vector in which values randomly give each entry (parameter) from real intervals. In this case, D represents the total number of parameters related to the chosen traveltime, where DE is applied. This first process is following described:

$$P[0][j][k] = frand(lowerPar[k], upperPar[k]), \quad \begin{array}{l} j \in \{1, ..., Np\} \\ k \in \{1, ..., \ D\} \end{array} , \tag{1}$$

Here, $P[0][j][k]$ specifies the j-th individual from the initial population and your k-th parameter. Besides, $frand(a, b)$ gives a random real value from $[a, b]$ interval, $lowerPar[k]$ holds the lower limitant and $upperPar[k]$ the upper limitant for the k-th traveltime parameter.

Right after this first step, the iterative process takes place. In each iteration i, pre-established rules called a mutation, crossover, and selection are sequentially applied over the $P[i-1]$ population to evolve it to the next generation $P[i]$. For these three rules, a detailed explanation is given below:

Mutation: For the mutation step, two (mutually different) random indexes $r_1, r_2 \in \{1, ..., NP\}$ are chosen based on a uniform integer distribution in order to produce a mutant population $V[i]$, as described by:

$$V[i][j][k] = P[i{-}1][j][k] \\ +F{*}(P[i{-}1][r_1][k]{-}P[i{-}1][r_2][k]) \quad \begin{matrix} j{\in}[1,Np] \\ \\ k{\in}[1,D] \end{matrix} \quad , \tag{2}$$

Here, one mutant member has generated for each j in the $P[i-1]$ population, using for that a mutation strategy very similar to the one referred to as "$DE/rand/1$", that imposes that every mutant parameter has to be composed by a linear combination of the randomly selected elements, being F a fixed constant called *scale factor*. Other mutation strategies are applicable, but, given the promising results produced by "DE/rand/1", they will not be considered in this work.

Crossover: Once the mutation part is finished, the crossover takes place, where a child population $U[i]$ comes to life with exactly Np individuals, such that each member j is composed partially by parameters from the population individual $P[i-1][j]$ and from the mutated individual $V[i][j]$, as described by:

$$U[i][j][k] = \begin{cases} V[i][j][k] & \text{if } (frand() \; < \; CR) \;\; or \;\; (k == k_{rand}) \\ P[i-1][j][k] & \text{otherwise.} \end{cases} , \tag{3}$$

In the above equation, CR is another fixed value called *crossover rate*, $frand()$ is a uniform random generator of real numbers in the real $[0,1]$ interval. Further, $k_{rand} \in \{1, ..., D\}$ is a random index chosen for each individual $j \in \{1, ..., Np\}$, ensuring that at least one parameter of the mutant individual j will be passed on to the child j.

Selection: This last step finishes the two previous ones. With it, the new generation $P[i]$ is created out of populations $P[i-1]$ and $U[i]$, in such a way that only best fitted individuals of each index $j \in \{1, ..., Np\}$ can survive for the next population. This selection process is following summarized:

$$P[i][j] = \begin{cases} U[i][j] & \text{if } f(P[i-1][j]) \; < \; f(U[i][j]), \\ P[i-1][j] & \text{otherwise.} \end{cases} , \tag{4}$$

where, $f(x)$ is a function that measures how much individual x fits in the environment, or more contextualized, how much the candidate set of parameters x is suited for the traveltime operator at a specific seismic data point.

Adaptive Differential Evolution (JADE):

Adaptive Differential Evolution (Zhang and Sanderson, 2009), or JADE, can be understood as a DE variant since initialization of population P is the same, and generation's stages are similar. However, three new changes are introduced: an adaptation of control parameters F and CR, another mutation strategy called "$DE/current-to-pbest/1$", and an optional archive population. Such modifications appear in the mutation and crossover stages, in which the F and CR parameters are not fixed anymore. Instead, they are generated at each iteration i for each individual j in the population $P[i-1]$, being respectively denoted by F_j and CR_j, and given by

$$F_j = cauchy(\mu_F, 0.1), \quad \text{and} \quad CR_j = normal(\mu_{CR}, 0.1). \tag{5}$$

In the above equation, $cauchy(\mu_F, 0.1)$ means the Cauchy distribution with location value 0.1 (fixed here for simplicity) and scale value μ_F. In the same way, $normal(\mu_{CR}, 0.1)$ means the normal distribution with standard deviation 0.1 (also fixed) and mean value μ_{CR}. The two parameters μ_F and μ_{CR} are initialized in the beginning and then updated at the end of each generation by:

$$\mu_F = 0.9 \cdot \mu_F \; + \; 0.1 \cdot mean_L(S_F), \tag{6}$$

$$\mu_{CR} = 0.9 \cdot \mu_{CR} \; + \; 0.1 \cdot mean_A(S_{CR}), \tag{7}$$

Here, S_F and S_{CR} denote the sets composed by the successful parameters F_j and CR_j, respectively, both associated with the $U[i][j]$ individual when the first statement of Eq. 4 is applied. Moreover, $mean_L(S_F)$ and $mean_A(S_{CR})$ designate the Lehmer mean and arithmetic average values of the sets S_F and S_{CR}, respectively. We recall that the Lehmer mean reads

$$mean_L = \frac{\sum\limits_{F \,\in\, S_F} F^2}{\sum\limits_{F \,\in\, S_F} F}, \tag{8}$$

Integrated with the above modifications, a new mutation strategy called $/de/current-to-pbest/1$ is used: for a given percentage number p, with $0 \leq p \leq 1$, the $100p\%$ best individuals are selected to generate the mutated individual $V[i][j]$ according to the expression

$$V[i][j] = P[i-1][j] + F_j * (P[i-1][j_{best}] - P[i-1][j]) + \\ F_j * (P[i-1][r_1] - P[i-1][r_2]) \tag{9}$$

Here, the three indexes $j_{best} \in [1, \lceil Np{\cdot}p \rceil]$, $r_1, r_2 \in [1, Np]$, are chosen according to an integer uniform distribution. The population is previously sorted so that the j_{best} individual, the one with the highest semblance value, occupies the position 1 and the remaining ones (until j_{best} position) occupy successive positions (in a decreasing fashion) according to the semblance value.

Remark: The population diversity, that is, the scheme that prevents the algorithm from being stuck in local maxima, can be improved upon the introduction of an archive population A, populated with the individuals from population $P[i-1]$ that failed on Eq. 4 to provide a semblance value greater than the one obtained by the $U[i]$ individuals. With the help of the archive A, the individual $P[i-1][r_2]$ in Eq. 9 has its r_2 index now ranging from the $[1, Np + |A|]$ interval, such that individual

Algorithm 1 Pseudo-code for JADE

```
 1: $sizeA = 0;$
 2: $pBest = \lceil p \times Np \rceil;$
 3:
 4: Create an empty population $A$
 5: Create a random initial population $P[0]$ of size Np
 6:
 7: // Iterates through all Ng generations
 8: for $i = 1$ to $Ng$ do
 9:     $meanSF = 0.0;\ meanSF2 = 0.0;$
10:     $meanSCR = 0.0;\ meanSCR2 = 0.0;$
11:
12:     // Finds the pBest individuals with the greatest semblance
13:     for $j = 1$ to $pBest$ do
14:         for $k = j + 1$ to $Np$ do
15:             if $f(P[i-1][k]) > f(P[i-1][j])$ then
16:                 Swap $P[i-1][k]$ with $P[i-1][j]$
17:
18:     // Updates each individual
19:     for $j = 1$ to $Np$ do
20:         do $F_j = min(1.0,\ cauchy(\mu_F, 0.1));$
21:         while $F_j \leq 0$
22:
23:         $CR_j = min(1.0,\ max(0.0,\ normal(\mu_{CR}, 0.1)));$
24:
25:         $u = P[i-1][j];$
26:         $x_1 = P[i-1][irand(1, Np)];$
27:         $x_{best} = P[i-1][irand(1, pBest)];$
28:         $r2 = irand(1, Np + sizeA);$
29:
30:         if $r2 \leq Np$ then $x_2 = P[i-1][r2];$
31:         else $x_2 = A[r2 - Np];$
32:
33:         $k_{rand} = irand(1, D)$
34:         for $k = 1$ to $D$ do
35:             if $k == k_{rand}\ or\ frand(0,1) < CR_j$ then
36:                 $u[k]\ \mathrel{+}=\ F_j \times (x_{best}[k] - u[k] + x_1[k] - x_2[k]);$
37:
38:                 // Verifies lower boundary
39:                 if $u[k] < lowerPar[k]$ then
40:                     $u[k] = (lowerPar[k] + P[i-1][j][k])/2;$
41:
42:                 // Verifies upper boundary
43:                 if $u[k] > upperPar[k]$ then
44:                     $u[k] = (upperPar[k] + P[i-1][j][k])/2;$
45:
46:         if $f(u) > f(P[i-1][j])$ then
47:             $P[i][j] = u;$
48:
49:             $meanSF \mathrel{+}= F_j;\ meanSF2 \mathrel{+}= F_j \times F_j;$
50:             $meanSCR \mathrel{+}= CR_j;\ meanSCR2 \mathrel{+}= 1.0;$
51:
52:             if $sizeA < Np$ then
53:                 $sizeA \mathrel{+}= 1;\ A[sizeA] = P[i-1][j];$
54:             else
55:                 $A[irand(1, sizeA)] = P[i-1][j];$
56:
57:     if $meanSCR2 > 0.0$ then
58:         $\mu_F = 0.9 \times \mu_F + 0.1 \times (meanSF2\ /\ meanSF);$
59:         $\mu_{CR} = 0.9 \times \mu_{CR} + 0.1 \times (meanSCR\ /\ meanSCR2);$
```

$P[i-1][r_2]$ is replaced by $\tilde{P}[r_2]$, according to

$$\tilde{P}[r_2] = \begin{cases} P[i-1][r_2] & \text{if } r_2 \leq Np, \\ A[r_2 - Np] & \text{otherwise.} \end{cases} \tag{10}$$

The maximum established size of the archive is set to be Np, and always when this limit is reached, an individual is randomly removed from its population, using for that a uniform integer distribution. The implementation of JADE is described in algorithm 1. In it, it is assumed a vector notation for the population. Further, mutation, crossover, and boundary verification are collapsed in one block (lines 31-39).

EXPERIMENTS

We considered two seismic datasets, DIFRAT and JEQUITI, being the first a synthetic and the second a real 2D dataset. Their specifications are displayed in Table 1. The midpoint and offset distances shown for the JEQUITI dataset are the mean value of all midpoint and offset distances since it is not a regular-grid dataset. Besides, JEQUITI is a marine two-dimensional (2D) line from the Jequitinhonha basin on the coast of Bahia, Brazil.

The estimation processes were executed on an Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz, 4 Cores with 15.6GB of CPU RAM, running with Ubuntu 16.04.4 and a graphics processing unit (GPU) GTX TITAN X, 12 GB VRAM. The OCT, ZO-CRS and FO-CRS code implementations, both for DE and JADE, were compiled with the *g++* compiler in its version *7.3.0* and *nvcc* in its version *9.0*, using on both the *-O3* optimization flag. Furthermore, the programs were implemented following the Scalable Partially Idempotent Task System (SPITS) programming model by Borin et al., 2016, then executed using the PY-PITS run time by (Benedicto et al., 2017).

QUALITATIVE RESULTS

For the given datasets, the parameter estimations provided by JADE and DE algorithms are compared and discussed. The cases for OCT, ZO-CRS, and FO-CRS are treated separately.

ZO-CRS:

From the semblance panels shown in Fig. 1, only a small difference between JADE and DE can be seen, mainly when considering Figs. 1c and 1d, which are pretty similar. Nevertheless, from results 1a and 1b, obtained with 11 individuals mutated during 11 generations, it is notable the better JADE convergence when compared with DE. However, as long as JADE and DE preserve the randomness in their very nature, this first good result in favor of JADE may be due to some lucky shot. Aiming at eliminating possible random effects, we performed multiple experiments, as can be seen in Fig. 3. In this result, each point represents the root mean square of a semblance panel obtained after Z in-

Table 1 Dataset specifications.

Name	Midpoint Distance (m)	Offset Distance (m)	Number of Traces	Number of time samples per trace	Sampling interval (ms)
DIFRAT	20	40	2626	376	4
JEQUITI	11.17	90.18	58189	1751	4

dividuals had been mutated during Z generations (being Z a value belonging to the x-axis range). The pattern al-ready obtained is confirmed, where JADE is performing better with the chosen operator and given dataset.

Also, the same repeats in Fig. 2 and 4 related with the real dataset, where it is still possible to see the difference in semblance intensity between panels 2a - 2b. Also, Fig. 4 confirms this difference in semblance intensity.

OCT:

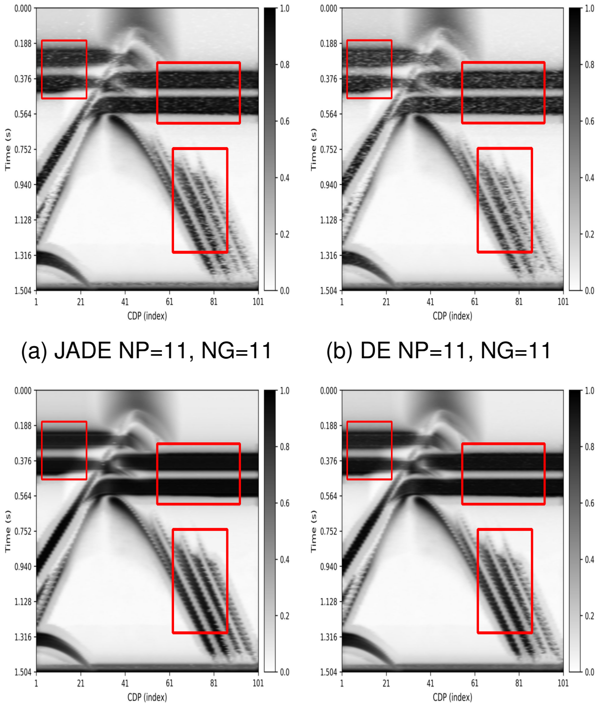
Now we present and analyze the OCT parameter-estimation results. From Fig. 5 it seems that JADE and DE had similar results. However, when a zoomed perspective is considered in parts of Fig. 5a, and Fig. 5b, where a 21 by 21 result is presented, the advantages of JADE over DE become apparent. In this case, DE had problems with convergence in multiples regions, whereas JADE overcame the convergence issues that DE had problems with. This result is directly confirmed by Fig. 7, where a similar convergence than that of Fig. 3 is obtained.

When considering the real dataset results in Fig. 9, a more in-depth look must be taken to identify the regions where JADE performed better. Despite the subtlety, it is possible to see that difference exists, with JADE having better or at least equal coherence in multiple points when comparing Fig. 9a and Fig. 9b. Also, Fig. 10 shows the resulting stacked section for three different offsets related to the real dataset. Finally, the difference is more significant in the convergence panel in Fig. 8.

FO-CRS:

The results from both JADE and DE for the FO-CRS case is analyzed now. Fig. 11 shows the sem-blance panel of two different offsets in DIFRAT dataset, while Fig. 13 shows these values for three other offsets in JEQUITI dataset.

Considering Fig. 11, notice that the increase in population and generation numbers improves the semblance value using both DE and JADE, but again, JADE provided a better convergence (Fig. 11c), where a 21×21 JADE is superior than the 21×21 DE (Fig. 11d). The same pattern repeats in Fig. 13, where a 21 by 21 DE (Fig. 13d) shows a worst result than a JADE (Fig. 13c). Also, Fig. 12 and 14 shows the differences in the stacking results, as well as Fig. 15 and 16 shows the convergence graphs for JADE and DE in DIFRAT and JEQUITI datasets, respectively.



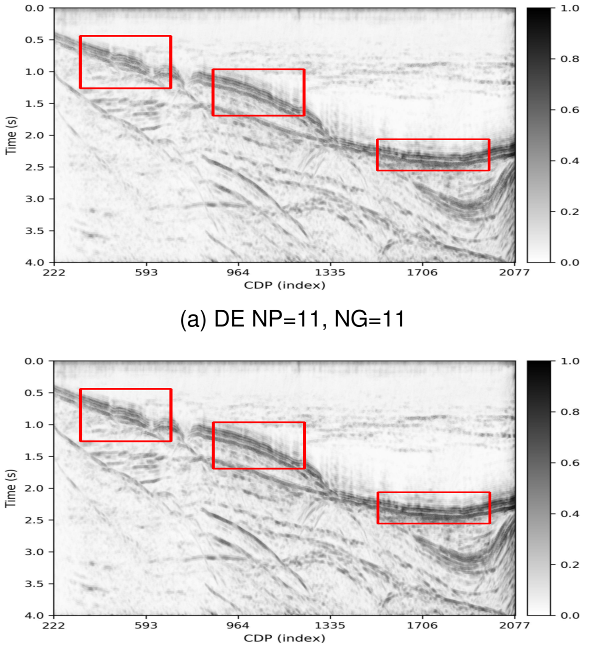
(a) JADE NP=11, NG=11

(b) DE NP=11, NG=11

(c) JADE NP=21, NG=21

(d) DE NP=21, NG=21

Figure 1. ZO-CRS semblance comparison between JADE and DE in DIFRAT synthetic dataset.



(a) DE NP=11, NG=11

(b) JADE NP=11, NG=11

Figure 2. ZO-CRS semblance comparison between JADE and DE in JEQUITI real dataset.

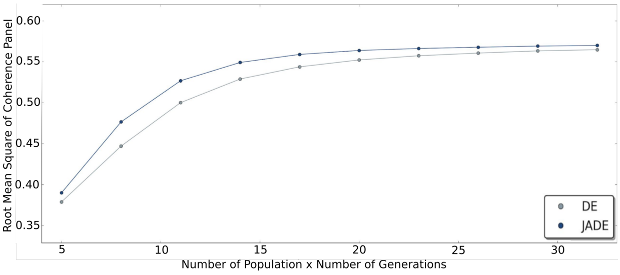


Figure 3. ZO-CRS convergence comparison between JADE and DE in DIFRAT synthetic dataset.

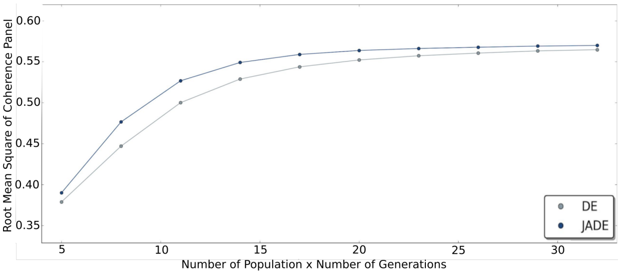
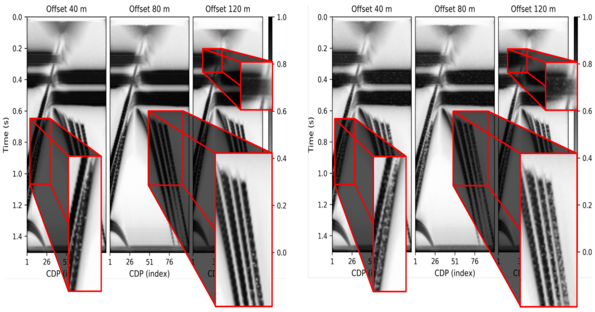


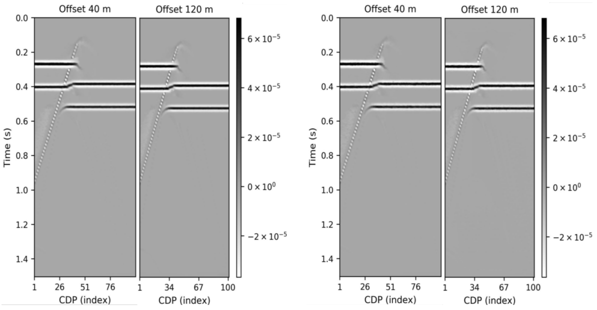
Figure 4. ZO-CRS convergence comparison between JADE and DE in JEQUITI real dataset.



(a) JADE NP=21, NG=21

(b) DE NP=21, NG=21

Figure 5. OCT semblance comparison between JADE and DE in DIFRAT synthetic dataset.



(a) JADE NP=21, NG=21

(b) DE NP=21, NG=21

Figure 6. OCT stack comparison between JADE and DE in Difrat synthetic dataset.

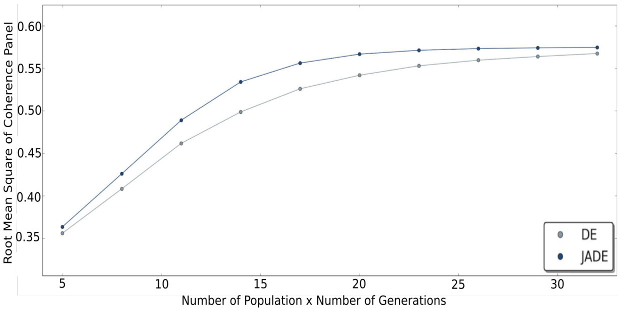


Figure 7. OCT convergence comparison between JADE and DE in DIFRAT synthetic dataset

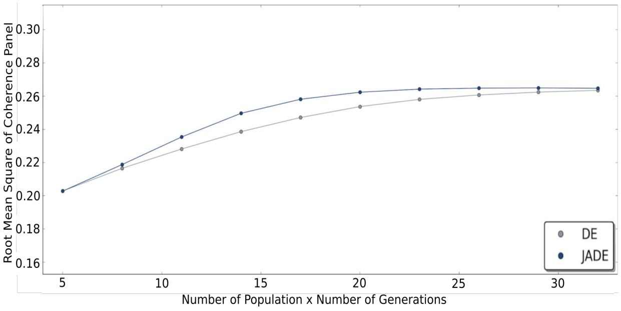
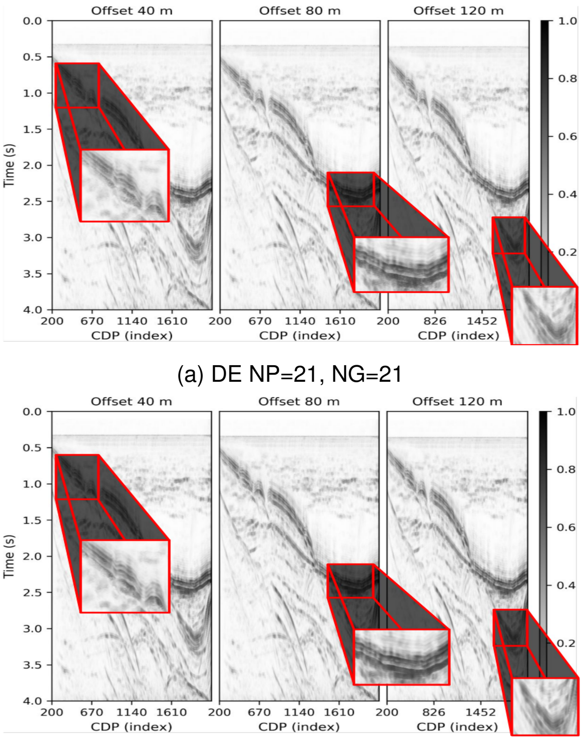


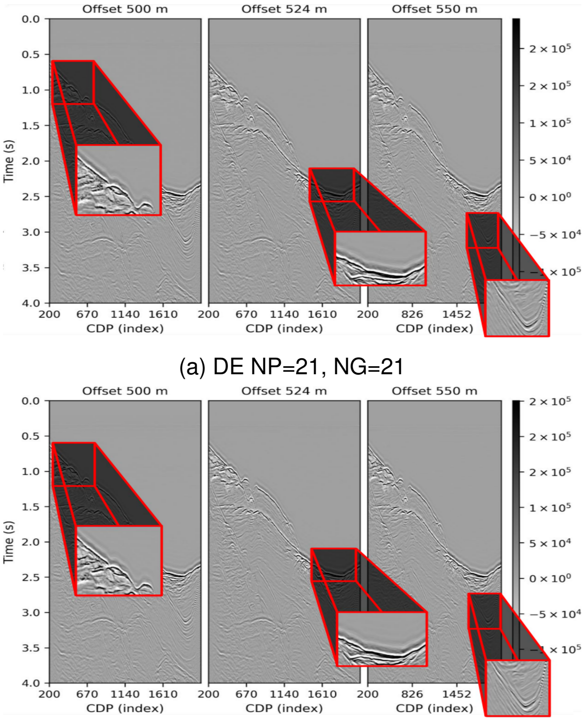
Figure 8. OCT convergence comparison between JADE and DE in JEQUITI real dataset.



(a) DE NP=21, NG=21

(b) JADE NP=21, NG=21

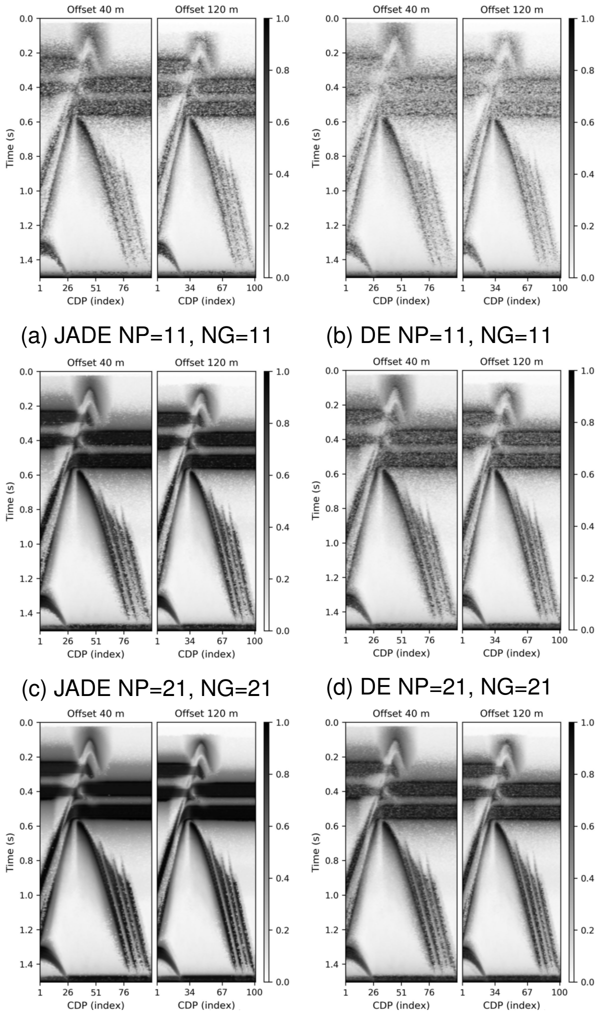
Figure 9. OCT semblance comparison between JADE and DE in JEQUITI real dataset.



(a) DE NP=21, NG=21

(b) JADE NP=21, NG=21

Figure 10. OCT stack comparison between JADE and DE in JEQUITI real dataset



(a) JADE NP=11, NG=11

(b) DE NP=11, NG=11

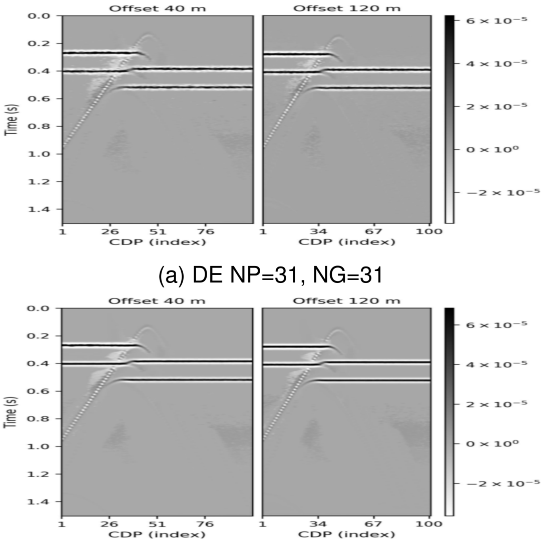
(c) JADE NP=21, NG=21

(d) DE NP=21, NG=21

(e) JADE NP=31, NG=31

(f) DE NP=31, NG=31

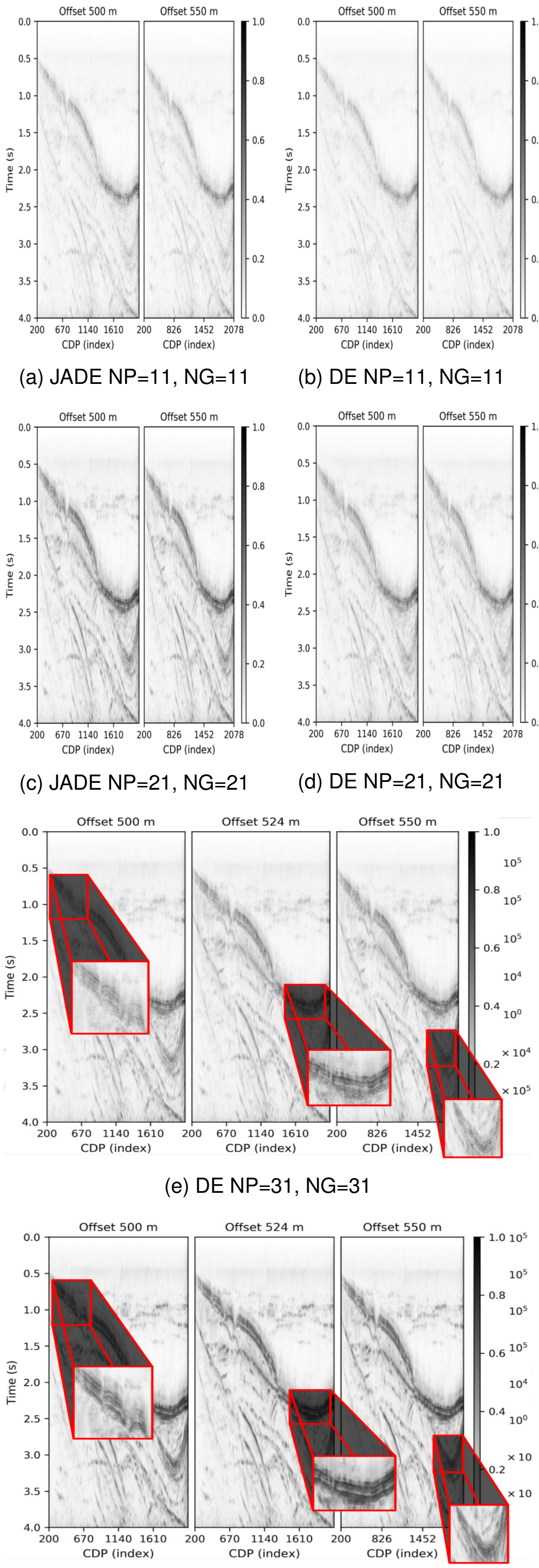
Figure 11. FO-CRS semblance comparison between JADE and DE in DIFRAT synthetic dataset



(a) DE NP=31, NG=31

(b) JADE NP=31, NG=31

Figure 12. FO-CRS stack comparison between JADE and DE in DIFRAT synthetic dataset.



(a) JADE NP=11, NG=11

(b) DE NP=11, NG=11

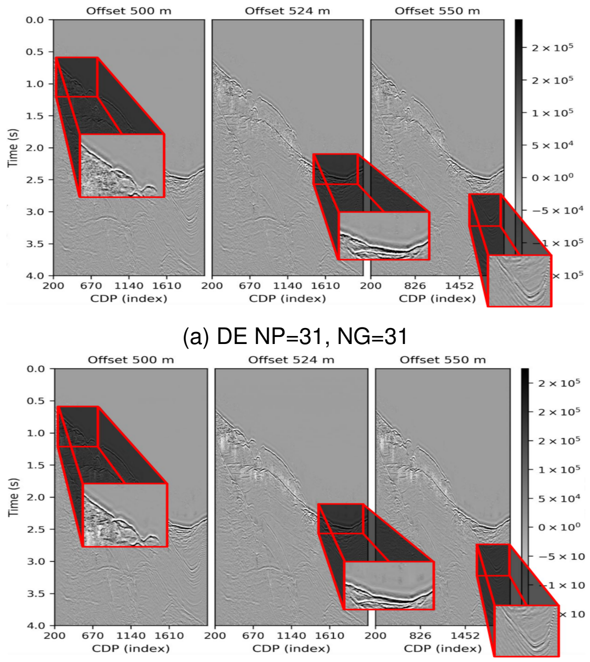
(c) JADE NP=21, NG=21

(d) DE NP=21, NG=21

(e) DE NP=31, NG=31

(f) JADE NP=31, NG=31

Figure 13. FO-CRS semblance comparison between JADE and DE in JEQUITI real dataset.



(a) DE NP=31, NG=31

(b) JADE NP=31, NG=31

Figure 14. FO-CRS stack comparison between JADE and DE in JEQUITI real dataset.

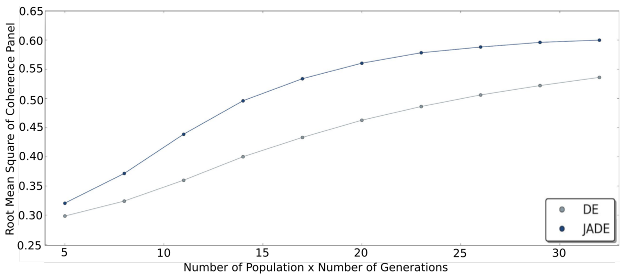


Figure 15. FO-CRS convergence comparison between JADE and DE in DIFRAT synthetic dataset.

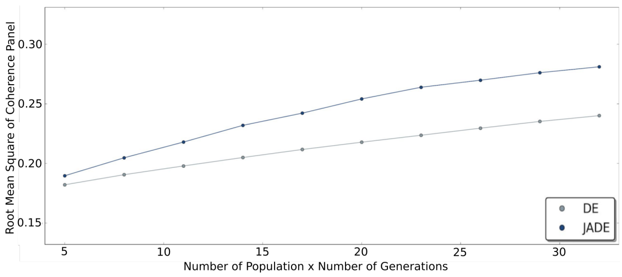


Figure 16. FO-CRS convergence comparison between JADE and DE in JEQUITI real dataset.

PERFORMANCE RESULTS

Analyzing and comparing both DE and JADE algorithms, it becomes clear that the population sorting (on lines 13-16 in Alg. 1) is the most significant part that could impact JADE performance since it is a simple selection sort with worst-case performance equal to $\mathcal{O}(n^2)$. However, as long as n is never greater than 10% of Np (in this work is a value fixed as $n/Np \leq 0.1$), this part becomes almost negligible when compared with all the other operations executed during the iterative process, mainly when the semblance calculation for each individual is considered, which is by far the most expensive section of both DE and JADE algorithms.

Therefore, even though JADE has elements that could slow it down at the point of losing its good convergence speed, the results from every experiment performed in this work have shown very similar execution times for both JADE and DE, revealing that the good results obtained by the adaptive algorithm must not be invalidated if performance time is being considered.

For instance, each parameter estimation was executed three times. The average time obtained from each set of three executions was computed and shown below. Tables 2, 4 and 6 presents the average execution time of each traveltime chosen, maintaining fixed the synthetic dataset DIFRAT. From those three tables, JADE performed $1.0528x$ slower than DE. However, when a bigger dataset is considered, JEQUITI, related with tables 3, 5 and 7, the performance difference between JADE and DE becomes almost negligible, being JADE only $1.0039x$ slower than DE. We conclude that not only JADE generated better results for the three cited traveltime and two selected datasets, but it also had almost the same performance as DE.

Table 2. ZO-CRS execution time: DIFRAT synthetic dataset.

Population	Number of	Execution Time		
Size	Generations	JADE (m)	DE (m)	JADE / DE (%)
11	11	0.01	0.01	86.88
21	21	0.03	0.03	111.56
31	31	0.04	0.04	113.99

Table 3. ZO-CRS execution time: JEQUITI real dataset.

Population	Number of	Execution Time		
Size	Generations	JADE (m)	DE (m)	JADE / DE (%)
11	11	0.42	0.40	104.02
21	21	1.09	1.05	103.15
31	31	2.40	2.34	102.84

Table 4. FO-CRS execution time: DIFRAT synthetic dataset.

Population	Number of	Execution Time		
Size	Generations	JADE (m)	DE (m)	JADE / DE (%)
11	11	0.03	0.03	109.34
21	21	0.09	0.08	109.07
31	31	0.14	0.12	111.03

Table 5. FO-CRS execution time: JEQUITI real dataset.

Population	Number of	Execution Time		
Size	Generations	JADE (m)	DE (m)	JADE / DE (%)
11	11	0.04	0.04	99.88
21	21	1.11	1.12	98.76
31	31	2.42	2.45	98.84

Table 6. OCT execution time: DIFRAT synthetic dataset.

Population	Number of	Execution Time		
Size	Generations	JADE (m)	DE (m)	JADE / DE (%)
11	11	0.15	0.14	102.80
21	21	0.36	0.35	101.90
31	31	0.58	0.58	100.99

Table 7. OCT execution time: JEQUITI real dataset.

Population	Number of	Execution Time		
Size	Generations	JADE (m)	DE (m)	JADE / DE (%)
11	11	2.32	2.31	100.61
21	21	6.50	6.53	99.57
31	31	13.83	14.43	95.86

DISCUSSIONS

Based on the results, it becomes clear the advantage of JADE over DE. However, it is still uncertain the reliability of those experiments, and once when dealing with metaheuristics, multiple factors can interfere in the quality of the results. Aiming to diminish this possible distrust, we discuss the expanded abstract presented at the 16th International Congress of the Brazilian Geophysical Society. In it, a completely different implementation was considered, where the central processing units (CPUs) were the main focus, with some parallelism being explored only with the SPITS structure (Benedicto et al., 2017). As opposed to this context, a new implementation using graphics processing units (GPUs) was glimpsed, and a new parallelism strategy had to be explored in order to obtain faster results in this work. Despite the differences, both implementations produced pretty similar and consistent quality and performance results.

Robustness is revealed by reaching similar results found in the literature (Barros et al., 2015, 2019) when using different implementations, starting from varying DE and JADE control parameters, like crossover rate (CR) and scale factor (F). For the OCT and ZO-CRS traveltimes, optimal CR and F are 0.9 and 0.8, respectively, while for the FO-CRS traveltime, both optimal control parameters are 0.5.

Contrary to the results obtained with DE, JADE has shown a more significant dependency on its control parameters (which are not few). The ones that interfered mostly in the quality of results were the presence of an archive population and the initialization of μ_F and μ_{CR} parameters. Regarding the archive population, it was noted that, in general, it helped improving convergence, but the maximum size of that population varied depending on the traveltime operator. For instance, a maximum size of Np for the archive was used when OCT and ZO-CRS traveltime was being used, and $Np/2$ when FO-CRS was considered. Also, the values that produced the best results for the control parameters μ_F and μ_{CR} were 0.5 and 0.7, respectively when FO-CRS was set as traveltime operator. When OCT was used, μ_F and μ_{CR} were set to 1.0 and 0.8. Furthermore, regarding the use of ZO-CRS, μ_F and μ_{CR} were fixed in 0.8 and 0.7.

From those two observations, it is impossible to guarantee the complete superiority of JADE over DE yet, mainly when considering the heuristic nature of both algorithms and the fact of JADE being more susceptible to be stuck on local maxima. However, from consistent results, JADE has been proving its value and can be a good alternative when convergence speed becomes an issue.

CONCLUSIONS

In this work, two metaheuristics were implemented to maximize the semblance objective function when estimating the parameters for seismic processing algorithms OCT, ZO-CRS, and FO-CRS, namely DE and JADE. It is well known that DE is a robust solution. However, its convergence can take many iterations, making the final results unsatisfying or computationally expensive. On the other hand, JADE could be used as an alternative, improving convergence time and quality. The results showed that both metaheuristics for the ZO-CRS program were able to converge properly, with slightly higher semblance values by JADE when using the same number of individuals and generations. A similar result could be noticed for OCT traveltime, with again JADE performing better than DE.

Nevertheless, in a more complex program, such as FO-CRS that needs to estimate five parameters, JADE offered considerably better convergence. Furthermore, for the same number of population individuals and generations, the program execution time remained close. Finally, it is clear that JADE is an acceptable alternative to the commonly used DE, given its faster convergence and time performance.

ACKNOWLEDGMENTS

This work was possible thanks to the support of Petrobras and Fapesp/Cepid #2013/08293-7-Brazil. The authors also thank the High-Performance Geophysics (HPG) team for technical support.

REFERENCES

Barros, T., Ferrari, R., Krummenauer, R., and Lopes, R. 2015. Differential evolution-based optimization procedure for automatic estimation of the common-reflection surface traveltime parameters. *Geophysics*, 80(6):WD189–WD200.

Barros, T., Lopes, R., Krummenauer, R., and Chauris, H. 2019. Common-offset common-reflection-surface attributes estimation with differential evolution. *Geophysical Prospecting*, 67(8):2022–2034.

Benedicto, C., Rodrigues, I. L., Tygel, M., Breternitz, M., and Borin, E. 2017. Harvesting the computational power of heterogeneous clusters to accelerate seismic processing. In *15th International Congress of the Brazilian Geophysical Society & EXPOGEF, Rio de Janeiro, Brazil, 31 July-3 August 2017*. Brazilian Geophysical Society.

Bonomi, E., Antonio M. Cristini1, D. Theis and Marchetti, P. 2009. 3D CRS analysis: a data-driven optimization for the simultaneous estimate of the eight parameters. In *SEG Technical Program Expanded Abstracts 2009*. Society of Exploration Geophysicists, Houston: SEG.

Borin, E., Benedicto, C., Rodrigues, I. L., Pisani, F., Tygel, M., and Breternitz, M. 2016. PY-PITS: A scalable Python runtime system for the computation of partially idempotent tasks. In *2016 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW), Los Angeles, CA, USA*, pages 7–12.

Coimbra, T. A., Novais, A., and Schleicher, J. 2016. Offset-continuation stacking: Theory and proof of concept. *Geophysics*, 81(5):V387–V401.

Garabito, G., Cruz, J. C., Hubral, P., and Costa, J. 2001. Common reflection surface stack: A new parameter search strategy by global optimization. In *SEG Technical Program Expanded Abstracts 2001*. Society of Exploration Geophysicists, San Antonio, Texas.

Jäger, R., Mann, J., Höcht, G., and Hubral, P. (2001). Common-reflection-surface stack: Image and attributes. *Geophysics*, 66(1):97–109.

Neidell, N. S. and Taner, M. T. (1971). Semblance and other coherency measures for multichannel data. *Geophysics*, 36(3):482–497.

Powell, M. J. D. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162.

Ribeiro, J., Okita, N., Coimbra, T., Ignácio, G., and Tygel, M. 2019. Using adaptive differential evolution algorithm to improve parameter estimation in seismic processing. In *Proceedings of the 16th International Congress of the Brazilian Geophysical Society&Expogef, Rio de Janeiro, Brazil, 19-22 August 2019*. Brazilian Geophysical Society.

Storn, R. and Price, K. 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359.

Walda, J. and Gajewski, D. 2017. Determination of wavefront attributes by differential evolution in the presence of conflicting dips. *Geophysics*, 82(4):V229–V239.

Zhang, J. and Sanderson, A. C. 2009. Jade: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5): 945–958.

Zhang, Y., Bergler, S., and Hubral, P. 2001. Common-reflection-surface (CRS) stack for common offset. *Geophysical Prospecting*, 49(6): 709–718.

Recebido em 4 de setembro de 2019. / Aceito em 11 de janeiro de 2021.

Received on September 4, 2019/ Accepted em January 11, 2021.